REMARKS

In response to the First Action Final Rejection dated November 27, 2006, and the Advisory Action dated March 22, 2006, we are continuing the application to allow the Examiner to perform a new search and give further consideration to the claims as stated in the Advisory Action. This Amendment includes the amendments presented in the Amendment filed after the first action final rejection as well as a clarifying amendment to claim 27 and a new independent claim 28.

As discussed, we believe that the cited Fuoco reference, United States Patent 5,452,429, has little or no teaching that corresponds to the currently pending claims.

As set forth in claim 1, the invention is a method that produces, from stored and regenerated parity check bits, a result that is used to directly **identify (i) the data buffer location that contains a data word with an erroneous bit, and (ii) the position of the erroneous bit in the data word contained in the identified data buffer location.** For clarity, we have changed the format of certain claims to highlight that two distinct things are identified by manipulation of the parity check bits.

We point out that the data words do not include the generated parity check bits. Further, we point out that the parity check bits are stored separately from the data words, that is, the parity check bits are stored in the buffer memory in multiple symbol storage locations that are separately addressable from the storage locations that store the data words. See, application page 10, lines 14-19 and Fig. 3. In contrast, Fuoco stores an

individual data code word and the corresponding generated parity bits (7 check bits and 4 bytes of data) in the *same* addressable memory location within the add-on memory. See, Fuoco Col. 3, lines 28-33; Col. 4, lines 6-9; Col. 4, lines. Thus Fuoco does not anticipate pending Claim 1 which specifically states:

> **storing the data words in a plurality of data buffer locations in the buffer memory and the parity check bits in one or more parity check buffer locations in the buffer memory, the parity check buffer locations being different locations than the data buffer locations that contain the data.**

For the same reasons, Fuoco does not anticipate any other of the independent claims or the claims that depend therefrom.

The current system does not, as is shown in Fuoco, use parity bits that are exclusive to a given data codeword to determine errors within that data codeword. Rather, the current invention utilizes the parity check bits that are associated with data that consists of multiple data words, to identify *both* the buffer location that contains an erroneous data word, and the location within the identified data word of an erroneous bit.

In Fuoco the parity bits and the corresponding data code word are read from a given memory location and the parity check bits are manipulated to identity an erroneous bit in the data word. See, e.g., Fuoco Col. 4, lines 29-31; Col. 5, lines 55 et seq; Col. 7, lines 60-64. In contrast, in the invention set forth in independent claims 1, 24 and 26 the parity bits and the multiple data words are read from multiple data and parity check buffer memory locations and the parity check bits are manipulated to identify two things, namely, (i) the data buffer location that contains a data word with an error, and (ii) which bit of the

identified data word is erroneous. Claim 1 states:

> **reading the stored data words and the parity check bits from the respective data buffer and parity check buffer locations;**
> **regenerating the parity check bits; and**
> **producing from the stored and regenerated parity check bits a result that is usable to directly identify**
> **i. the data buffer location that contains a data word with an erroneous bit, and**
> **ii. the position of the erroneous bit in the data word contained in the identified data buffer location.**

The Fuoco system, like conventional encoding and decoding systems, identifies the location of the erroneous bit within a given data codeword using parity check bits that are exclusively contained in that data codeword. Thus, the Fuoco system does not anticipate independent claim 1 and for the same reasons the remaining independent claims and the claims that depend therefrom.

While the Examiner quotes at length a description of how the data and associated parity check bits are entered into the particular memories used in the Fuoco system, the claims of the current application are not directed to overwriting all or portions of a data codeword contained in memory, rather they are directed to a method of encoding or decoding data. Accordingly, the fact that the Fuoco system uses a particular type of memory with particular lines that are asserted or deasserted to over write all or portions of a stored data codeword is of little consequence, since the encoding and decoding mechanisms used in Fuoco do not teach or suggest the method of encoding or decoding set forth in the pending claims.

Indeed, the Fuoco reference describes how the parity bits are re-generated for a given data word when the data word is only partially overwritten, that is, when fewer than the four data bytes are overwritten. The Fuoco performs a read-modify-write operation in which the encoding involved in the re-generation of the parity bits utilizes only the bytes of the single data word. See, Fuoco Col. 7, line 41- Col. 8, line 19. Thus, there is no teaching or suggestion in the read-write-modify operations of Fuoco of the current invention of claims 23, 25 and 27, which set forth that *multiple* data words are encoded to produce the parity check bits.

Further, we again point out that the invention of claim 23 is a particular method of encoding that uses a generator matrix that has a parity check generation portion that comprises **rows of bits that correspond to binary representations of the respective buffer locations that are used to store the data words**. There is no teaching or suggestion of such an encoding method in the Fuoco reference. Again, the mechanism by which Fuoco overwrites all or portions of a particular data codeword that consists of 4 bytes of data and their corresponding 7 parity check bits has little to do with the encoding method that is the subject of claim 23. For example, the Fuoco system does not supply *multiple* data words (in the example 1024 data words, see page 10, lines 16-17) to a generator matrix to produce the parity check bits.

The system claim of 25 similarly includes the generator matrix that is discussed above with respect to claim 23. The Fuoco reference does not teach such a generator matrix, but instead teaches its own particular syndrome generation mechanism that differs

considerably from the mechanism used in the current system. The only commonality between the two systems is that each uses a generator matrix. However, the generator matrices of the two systems are completely different, as are the objectives of the encoding schemes of the current invention and the Fuoco reference. For example, Fuoco does not teach a decoding operation that identifies a buffer location that contains a data word with an erroneous bit *in addition* to the location of the erroneous bit.

The Fuoco encoding mechanism is designed such that particular data bytes and their associated parity check bits, which together form a data codeword, are stored in a given memory location and can be selectively either completely overwritten or partially overwritten, in read-modify-write operations. Accordingly, the encoding mechanism of Fuoco and/or how Fuoco writes data bytes and parity check bits to a given memory location by asserting particular lines and so forth do not show, teach or suggest the system of claim 23. Specifically, Fuoco does not show, teach or suggest a controller that is operable to apply data to a generator matrix that includes a parity check generation portion, with rows of bits corresponding to binary representations of the respective data buffer locations to be used to store the data, as set forth in independent claims 23. For the same reasons, the apparatus of claims 26 and 27 are also not taught or suggested by the Fuoco system because, *inter alia*, the Fuoco reference does not teach or suggest the use of the parity generator matrix set forth in those claims.

Further, there is no teaching or suggestion in Fuoco of the invention as set forth in dependent claims 8 and 11. Fuoco does not teach or suggest producing a parity check bit

that corresponds to the parity check bits generated for the data. Fuoco teaches generating parity check bits only for the data bytes in a given data code word - see Table 1 of Fuoco.

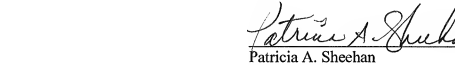
We do not specifically address the Examiner's rejections of certain of the dependent claims. This should not be construed as acquiescence to the rejections, but as recognition that the rejections are moot based on our remarks regarding the allowability of the independent claims.

New claim 28 is allowable for the same reasons discussed above.

The claims, as amended for clarity, should now be in form for allowance, and we request that the Examiner reconsider the rejections and issue a Notice of Allowance for all pending claims.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,



Patricia A. Sheehan
Reg. No. 32,301
CESARI AND MCKENNA, LLP
88 Black Falcon Avenue
Boston, MA 02210-2414
(617) 951-2500